



Adaptive Guidance based on Context Profile for Software Process Modeling

Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Oussalah

► To cite this version:

Hamid Khemissa, Mohamed Ahmed-Nacer, Mourad Oussalah. Adaptive Guidance based on Context Profile for Software Process Modeling. International Journal of Computer Science and Information Technologies, 2012, pp.50-60. 10.5815/ijitcs.2012.07.07 . hal-01139326

HAL Id: hal-01139326

<https://hal.science/hal-01139326>

Submitted on 28 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Guidance based on Context Profile for Software Process Modeling

Hamid Khemissa, Mohamed Ahmed-Nacer

Computer Systems Laboratory, Computer Science Institute, USTHB University,
El Alia BP n°32, Bab Ezzouar Algeria. Tel/Fax (00)213 21247917
hkhemissa@usthb.dz, hkhemissa@hotmail.com, anacer@cerist.dz

Mourad Ouassalah

Computer Laboratory Nantes Atlantique, Faculty of science, Nantes University
02 rue de la Houssiniere- BP 92208 44322 nantes Cedex 3. Tel 0251125847, Fax 0251125858
Mourad.Ouassalah@univ-nantes.fr

Abstract—This paper aims to define an adaptive guidance for software process modeling. The proposed guidance approach is based on development's profile context (actor's role in the process, actor's qualification and related activities in progress). We introduce new guidance concepts through adaptive guidance meta-model (AGM) allowing specific assistance interventions (corrective, constructive and automatic guidance). We illustrate our guidance approach using SPEM formalism extended with these new guidance concepts.

Index Terms—Meta Modeling, Software Process Modeling, Guidance Profile, Adaptive Guidance, SPEM Extension

I. Introduction

To attempt a better software quality and keep consistency and productivity aspects in software development, requires assisting developers at both organizational and methodological levels as well as on the software product consistency [1][2][3]. A rigorous assistance targets two aspects: 1) the control progress of the software process development regarding the temporal constraints of activity and the consistency of the results, and 2) the guidance adapted to the specific needs within the context of the activity in progress.

A guidance model to the software engineering should gather the essential characteristics making it possible to build the assistance system and to graft it with the development system. Several process-centered environments [4][5][6][7] deal with the assistance aspect in the support of the software product development. The execution software of these environments guides and supports the user in performing the software process activities. However, the provided guidance is often defined to be not adapted to a development profile context. The guidance

orientations are defined on the basis that the human actor, regardless his profile, has a central role in the progress of the development process.

The principle of the proposed approach is to consider that a process of software development is entirely based on collaboration between performers, who perform activities to reach a common goal called software product. Each performer supports the achievement of a particular task. This work requires a refined guidance adapted to the performer's profile. This profile is described respectively by the task performer model (defined by its role, qualification and behavior) and the activity model, in relation to the task's context in progress. The guidance system must be able to agree with the activity in all its complexity and to provide an adaptive guidance related to task progression.

Our approach uses SPEM [8] (Software and System Process Engineering Meta model Specification) that is considered as the reference Meta model providing basic concepts necessary to model various software processes. So, the proposed approach defines concepts and principles of adaptive guidance, these will be used to extend SPEM Meta model in the form of stereotypes relating to guidance in the packages' profile "MethodContent" and "ProcessWithMethod" as well as their instantiation, respectively, by new stereotyped classes and associations.

Our modeling approach denoted AGM (Adaptive Guidance Meta Model) is based on the Adaptive Concepts regarding to the development current context. Its adaptation considers explicitly the activity and task abstraction. These two abstraction levels are described by four basic models related to the activity, activity performer, task and task performer models interlinked by the respective associations.

Section 2 of this paper presents a synthesis of similar work and describes the current trend. Section 3 presents our AGM modeling approach while section 4 describes the guidance concepts integration in the SPEM Meta

Model. Section 5 presents a practical interpretation of our approach and section 6 concludes this paper with some future works.

II. Current Trend

Several process-centered environments [7][9][10] deal with the assistance aspect in the support of the software product development. Some Process-Centered Software Engineering Environments (PSEE) use an assistance description structured in phases like prescribing systems or proactive systems to control the operations carried out by the developer. However, they are essentially limited to the adaptive assistance aspect to current context of development. Taking into account specific criteria for adaptive guidance, these limits are classified into two categories: the first describes the basic concepts explicitly linked to the adaptive guidance aspect and the second category specifies the considered abstraction levels and the availability of a Specific Process Modeling Language.

In the first category:

- ▶ Global guidance core: The basic guidance is defined as a global orientation core regardless the profile of both the activity and the performer.
- ▶ Human performer profile oriented guidance: the guidance orientations are defined on the basis that the human actor, regardless his profile, has a central role in the progress of the development process.
- ▶ Context development guidance: The selection of the appropriate type of guidance is more often not adapted nor suitable to a current context.
- ▶ Guidance types: the selection of guidance types remains defined in a manual and in an intuitive way. It depends on the experience and on the informal personality of the project manager.

In the second category:

- ▶ Explicit activity and task abstractions : the guidance is often defined without explicit consideration of the fundamental abstractions of software process modeling; task and activity.
- ▶ Process Modeling Language (PML): the development environment often does not have a specific process modeling language. PML is defined in a preference order by explicit, predefined or implicit primitives.

To respond to these limits, one currently tries to offer more flexibility in the language of software process modeling. This tendency results in the idea to define

interventions of direct and adaptive assistance in particular contexts during the progress of software process. Among this new generation of PSEEs, we will give a comparative study, based on the considered criteria, of the following Meta-models: APEL [5] and SPEM [8] as the most representative in the software process modeling, RHODES [6][10] and ADDD [4] as they use basic concepts nearest to those introduced by our approach.

ADELE/APEL is based on a reactive database. It proposes a global assistance of proscriptive type without considering the performer profile and automates part of the development process using triggers. Its main purpose is the support of interoperability of software process.

SPEM Meta model introduced the concept of "Guidance". According to SPEM, the Guidance is a describable element which provides additional information to define the describable elements of modeling. SPEM does not develop the guidance concept. Its definition and practice across the various packages is not approached in details. However, it offers, through the stereotype "Guidance_kind" different types of guidance such as: Template, Guidelines, Checklists, etc.... But, SPEM does not offer models of preset guidance nor directives of uses and selection of Guidances_Kind. The selection of guidance types remains defined in a manual and in an intuitive way. It depends on the experience and on the informal personality of the project manager. In addition, the proposed guidance is not suitable to the performer's profile (role, qualifications and behavior).

RHODES/PBOOL+ uses a strategy model and an explicit description of a development process. The software processes are modeled in PBOOL language. The activities are associated to a guidance system with various scenarios of possible realization.

ADDD/ALADYN provides process automation and controls the impact in a concrete system. The task hierarchy is used to organize the descriptions of a process, called policies. Several aspects are grouped and treated in a policy (parameters and procedures used for customization and triggers used for process supervision, reporting, automation and control consistency). A policy can be instantiated for several tasks. The instantiated triggers are rules of the form event-condition-action (ECA) and used to implement a reactive behavior.

A comparative table of the studied Meta models (see Table-1-) is described as follows:

Table -1- : A comparative table of the studied Meta models

Meta model		ADELE /APEL	RHODES / PBOOL+	ADDD / ALADYN	SPEM
Criteria					
First Category	Global guidance core	Global	Global	Customized for each task	Global
	Human performer profile oriented guidance	Not adapted	considered strategy Model	Not adapted	Not adapted
	Context development Guidance	Not adapted	Adapted	Adapted	Not adapted
	Guidance types	Not invoked	associated with a specific guide system	Not invoked	Intuitive selection
Second Category	Explicit activity abstraction	Explicit abstraction	Implicit abstraction	Implicit abstraction	Explicit abstraction
	Explicit task abstraction	Implicit abstraction	Not invoked	Explicit abstraction	Explicit abstraction
	Process Modeling Language(PML)	APEL With predefined primitives	PBOOL+ With explicit primitives	ALADYN Not explicitly mentioned	UML Profile With explicit primitive

The current tendency is that performers would like to have integrated environments that are suitable to specific needs according to the role and the characteristics of each performer. However, the provided efforts to develop such environments remain an insufficient contribution. This generation of guidance environment still interests researchers in defining new concepts and objectives of the software process modeling [2][3][11][12][13].

Our work proposes an approach to define adaptive guidance modeling in software process. The proposed approach concepts are described through a Meta model. The information provided must be adapted to the development context profile. They must guide the performer during the software process development through suitable actions and decisions to undertake with corrective, constructive or automatic intervention [3][11].

III. The Adaptive Guidance Meta-model(AGM)

Our modeling approach AGM is defined considering the identified limitations of studied PSEEs. Among the essential characteristics of our approach is to consider the current context adaptation aspect, described by the attributes of our different models which are structured on two abstraction levels.

In this context, our conceptual Meta model is based on the typical reasoning of software processes enriched by the adaptive guidance element. This one controls the good progress of activities and offers an adaptive guidance to the performer. It is described schematically as follows:

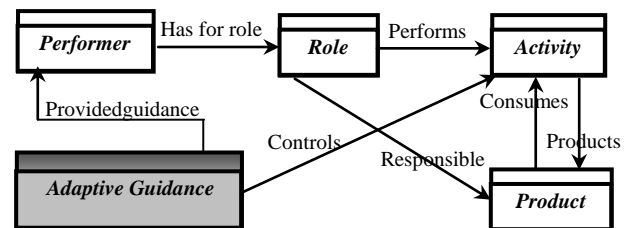


Figure-1- : The adaptive guidance relationship in software process modeling

To define an adaptive guidance model to the conceptual context, our approach considers two levels of abstraction, the activity and task levels. This explicit definition highlights the links between the abstraction concepts of both levels based on the four introduced models. This is in order to better differentiate and structure the adaptive guidance interventions.

In our approach, the activity concept is described by the set of activity and task models inter-related by the respective associations; as well as the concept Performer is represented by the set of activity performer and task performer models inter-related by the respective associations.

This new description defines an adaptive guidance Meta model based on the activity and task models related respectively by the association class "ProcessRole" to the corresponding activity performer and task performer models. To ensure consistency between these four models, the activity and the activity performer models are respectively defined as a set of occurrences of structured and organized task and task performer models which respectively inherit the

characteristics of the activity performer and the activity models.

The proposed Meta model aims to generate the assistance interventions adapted to the development context related with the considered specific properties and data of each defined models (See Figure-2-).

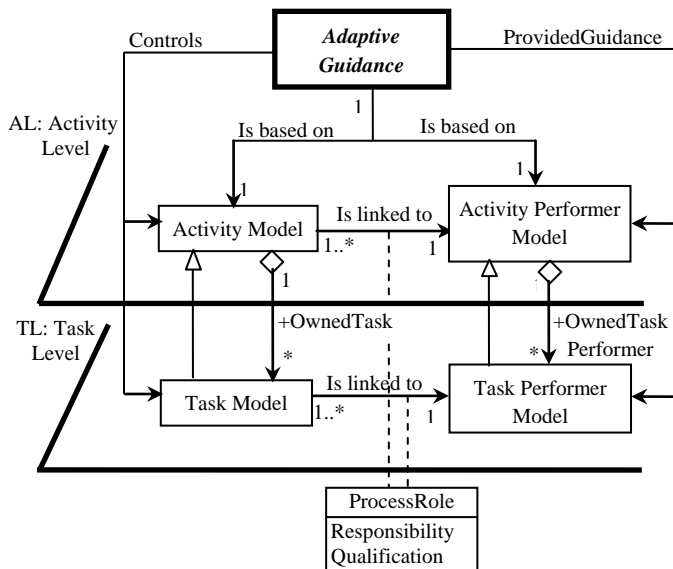


Figure-2- : The adaptive guidance conceptual Meta model.

3.1 The adaptive guidance Meta model.

The description of the four introduced models regarding the two abstraction levels is given as follows.

3.1.1 At the Activity level

1. **The activity model:** models the workflow, it is defined by:

- A hierarchical list of tasks,
- A mode of progression in the activity ensuring that all tasks can be performed under control in a preset order established by the designer,
- A temporal mode of progression specifying deadlines for completion.

The aspects of the activity model are useful for the assistance system to provide guidance on contextual growth in activity.

2. **The activity performer model:** development environments allow exchanges and collaborative work. The guidance system can then construct an activity performer model that represents the team's elements. Example: trace of the various activities of the team as well as different interactions allow the performer to have a script about his own progress in the activity and the progression of the activity. The properties of this model can be static or dynamic order.

- The static dimension referencing skills and performer performance in the field of collaboration and task distribution.
- The dynamic dimension deals with the activity performer behavior. It describes the actions taken by the performer during the course of software process.

These data constitute indication that can be interpreted on the use of the assistance by the performer.

3.1.2 At the Task level

1. **The task model:** defines the current context of task. It allows our model to make adaptation according to this specific context.

- An associate process performer.
- List of source and target products.
- A temporal mode of progression specifying deadline for completion.

2. **The task performer model:** defines the specific properties of each performer. It allows our model to make adaptation according to these properties while maintaining the activity model. These properties can be either static or dynamic.

- The static aspect refers to the user characteristics:
 - his expertise in the field,
 - his familiarity with the software process model or with the software process,
 - his role in the activity.
- The dynamic aspect refers to the behavior of using the guidance system, which must be interpreted during the use of the process or guidance system, for example:
 - the fact to execute, define or complete the resource of software process,
 - the workload of an activity,
 - his reaction to a support message.

3.2 The assistance intervention

During the construction or interpretation of a software process model, the proposed guidance model allows the performer to choose various support functions, namely:

- **Control and taking corrective initiative:** protect the user of his own initiatives when they are inadequate under progress.
- **Control and taking constructive initiative:** the ability to take positive initiatives, executing and combining the performance of operations without the user intervention.
- **Automatic assistance:** analyze the impact projection to avoid deadlocks or delays.

3.3 AGM Characteristics

Based on the detailed description of our modeling attributes, our AGM Meta model compared to the considered criteria is described at the following table.

Table -2- : The Adaptive Guidance Meta model characteristics

Meta model		AGM	Signification
Criteria			
First Category	Global guidance core	Specific for each task and activity	AGM provides specific assistance to each activity and task.
	Human performer profile oriented guidance	Adapted to the performer's profile	Any specific guidance provided is adapted to support the appropriate performer's profile.
	Context development Guidance	Adapted of the development context	Any specific guidance is adapted to the development context profile based on the performer, task and activity models.
	Guidance types	Provide a corrective, constructive or automatic guidance.	Our adaptive guidance must guide the performer during the software process development through suitable actions and decisions to undertake with corrective, constructive or automatic intervention.
Second Category	Explicit task and activity abstraction	Explicit abstraction	The guidance adaptation is based on the explicit task and activity abstractions.
	Process Modeling Language(PML)	UML Profile	We use the UML profile to the software process modeling with explicit primitives inherited from the UML language.

At the first criteria category, consideration of the basic concepts in our Meta Model represents a support which targets the adaptation and the guidance profile.

The guidance adaptation is defined according to the abstraction level considering the task or the activity performer profile. This is explicitly described by "Linked to Activity Performer Model" and "Linked to Task Performer Model" association.

The second criteria category emphasizes our modeling approach at the abstraction aspect level and the consideration of process model language with explicit primitives.

IV. SPEM Extension

The adaptive assistance concepts proposed by our approach are dedicated to any system of software process modeling. As SPEM is a reference Meta model in the software process modeling, we considered useful to illustrate our approach by extending SPEM with these new concepts. Respecting the synoptic elaboration of SPEM [8], this extension will be done explicitly, by defining the adaptive guidance concepts, at the package level: "MethodContent" and "ProcessWithMethod". So, we introduce and illustrate in details the description of the main stereotypes "GuidanceDefinition" and "GuidanceUse" respectively in the package level "MethodContent" and "ProcessWithMethod" in order to allow the exploitation of predefined guidance and describe the use of guidance according to the current context.

4.1. The "MethodContent" package with adaptive guidance

The "MethodContext" package defines the basic element of each method such as Role, Task and Work Product. Most classes and stereotypes of this package have the particularity of having the suffix "Definition" in order to express the definition of the element. Respecting this logic to extend the profile of the package, we have defined the new stereotype "GuidanceDefinition" that defines any guidance which can be used in a development approach (see Figure -3-).

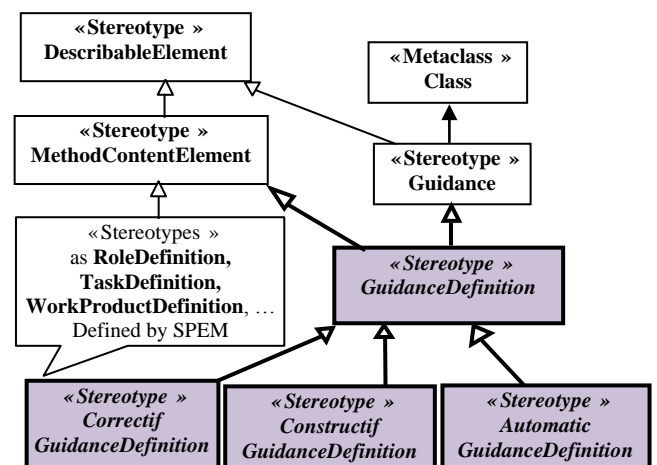


Figure-3-: Extending SPEM2.0 profile with the "GuidanceDefinition" stereotype.

The stereotyped class "GuidanceDefinition" in relation with the basic elements and their relations with the subclasses of package elements "MethodContent" are given in details in a diagram (see Figure-4-), and

by the description of its association and semantic definition. The relations are defined to describe how to perform the method according to the modeler perception.

Taking into consideration the performer profile, the "GuidanceDefinition" instance expresses that any guidance definition is made according to the task instant context to realize. A document "GuidanceDefinition" represents the guidance offered by the instances of a role definition and / or the required guidance for the execution of a task. The definition of the performer's role and the task context are presumed to be useful to find the most adapted guidance during the interpretation of the predefined method content and dynamically assign the most appropriate guidance. The "GuidanceDefinition" class inherits the guidance concepts defined in SPEM.

This new stereotyped class is described as an element of "MethodContent" providing additional information related to describable elements and defines recognized guidance in the software development domain. It describes the appropriate guidance to a "TaskDefinition" instance taking into account the provided competences by the class "Default-TaskDefinitionPerformer" associated to a defined role. This dependency is reflected by the following associations:

- "providedGuidance-T: GuidanceDefinition" expresses the fact that "GuidanceDefinition" instance provides a lot of defined guidance compared with the characteristics of the task performer "Default-TaskDefinitionPerformer" linked to this "RoleDefinition" and "TaskDefinition" classes. i.e.: the task performer may have zero or many guidance definitions as an adapted support to the task performer profile.
- "providedGuidance-A: GuidanceDefinition" expresses the fact that "GuidanceDefinition" instance provides a lot of defined guidance compared with the characteristics of the activity performer "Default-ResponsibilityAssignment" linked to this "RoleDefinition" and "WorkProductDefinition" classes. i.e.: the activity performer may have zero or many guidance definitions as an adapted guidance to his profile.
- "controlledtask: Task Definition" expresses the fact that "TaskDefinition" instance may be controlled by zero or many adapted guidance, and a guidance definition can be offered at zero to many tasks.

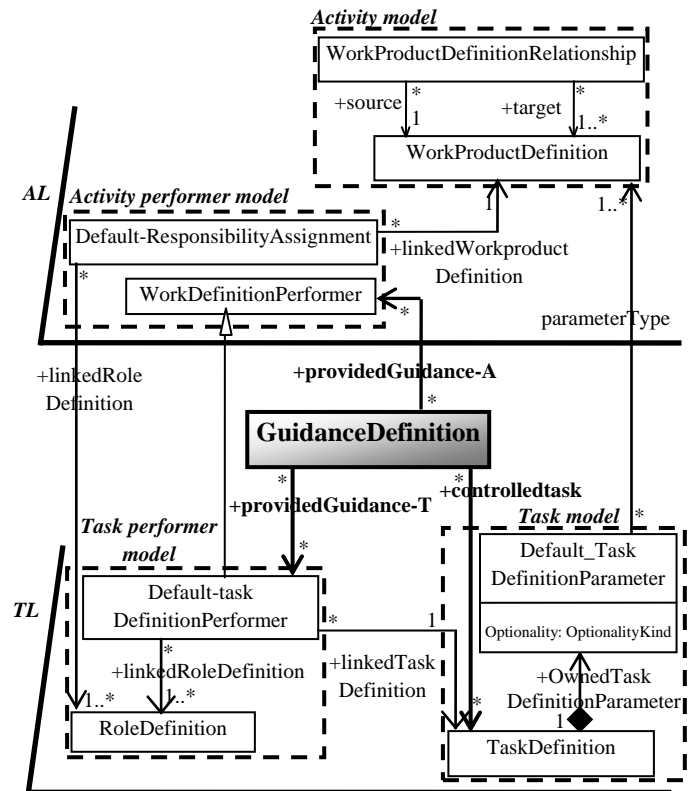


Figure-4 : The main classes and relations of extended package "MethodContent"

4.2. The "ProcessWithMethod" package

This package describes the use of the development elements and methods describing a given software process in accordance with the "ProcessStructure" concepts. The Method content use is an abstract generalization for special breakdown elements that references one concrete method content element. The product of this description is a software process described in a particular context and according to the retained life cycle. Most classes and stereotypes of this package have the suffix "Use". The Use concept means the use of an instance class on a specific case. Thus, we extend SPEM profile with new stereotype "GuidanceUse" and its different guidance categories (see Figure -5-).

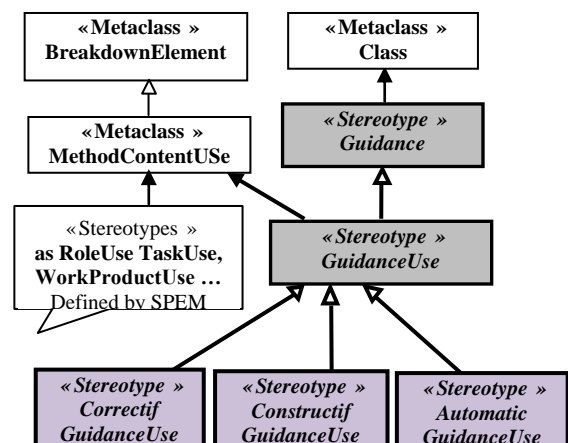


Figure -5- : Extended SPEM 2.0 Profile by the "GuidanceUse" stereotype

We define the guidance interventions for particular uses in "ProcessWithMethod" through the class "GuidanceUse". This stereotype role will allow the description of the guidance used in particular contexts. Its structural relation and its proper association sets are described in Figure -6-.

The «*GuidanceUse*» is a key concept to insure the application of the defined guidance "GuidanceDefinition". It is considered as a reference object for an adapted guidance to a particular context. The use of guidance depends on the description and the evolution of the "TaskUse" element as well as the role and the qualifications of the performer applied by "RoleUse" element. The "GuidanceUse" class inherits the guidance concepts defined in SPEM.

This new stereotyped class is described as a process element. It explicit the adaptive use of the defined guidance in relation to a particular context. It takes into account the characteristics of a "TaskUse" as well as the qualifications and performer behavior

"ProcessPerformer". This dependency is reflected by the following associations:

- *guidance*: *GuidanceDefinition*; represents the link between one or more elements "GuidanceUse" to an element of "GuidanceDefinition". A "GuidanceDefinition" class can be represented by many "GuidanceUse" classes.
- *providedguidance*: *GuidanceUse*; expresses the fact that "GuidanceUse" element provides many defined guidances for the "ProcessPerformer" or "ProcessResponsibilityAssignment" instance linked to its "roleUse". And the performer may have zero or more "GuidanceUse" in relation with his profile.
- *controlledtask*: *TaskUse*; the "TaskUse" instance may be assisted by zero to several guidance, and a "GuidanceUse" instance may be provided at zero or more tasks.

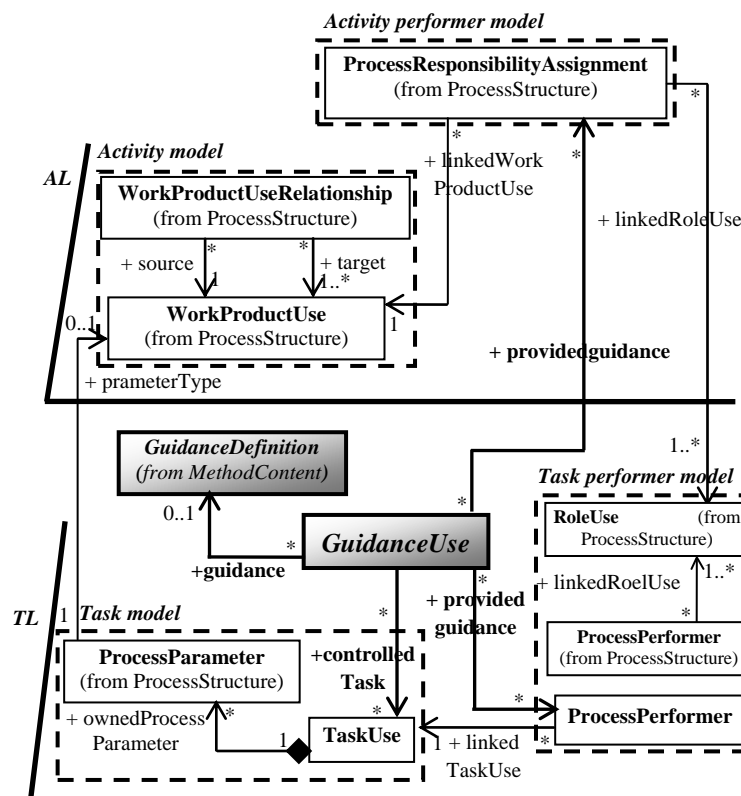


Figure -6- : The main classes and relations of extended package "ProcessWithMethod"

V. Practical Interpretation of Our Approach

Considering the software process model "Activity test", the process "Activity test" in the software development is composed of several types of tests such as: Integration test and Unitary test. Each receives as input a test plan and provides a test report. For each

type of test, there is a manager, responsible of the execution.

The activity process "Activity test" is described by a performing tree given in Figure -7-. We notice that the activity test starts the execution of subactivities "Unitary test" then "Integration test". The unitary test launches in parallel the execution of tasks "Test unit".

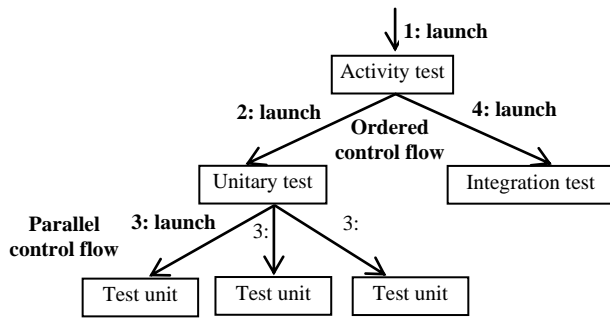


Figure 7- : The process of the activity test

To simplify our example, we consider the case where the execution of test is in the unitary component test. The application of the activity "Unitary test", requires the list of components. It calls the tool that will create the necessary environment to carry out the actual execution of the "Unitary test", as the state diagram, the test variable, etc. ... the activity "Unitary test" launches in parallel the different tasks "Test unit" and an event signals the beginning of the "Test unit" execution. Finally, the ended event is broadcast.

Concerning our approach, the adaptive execution process of the activity "Unitary test", regarding our Adaptive Guidance Meta model, is described by Figure-8-. I did not label the associations since they are practically the same as the Meta model.

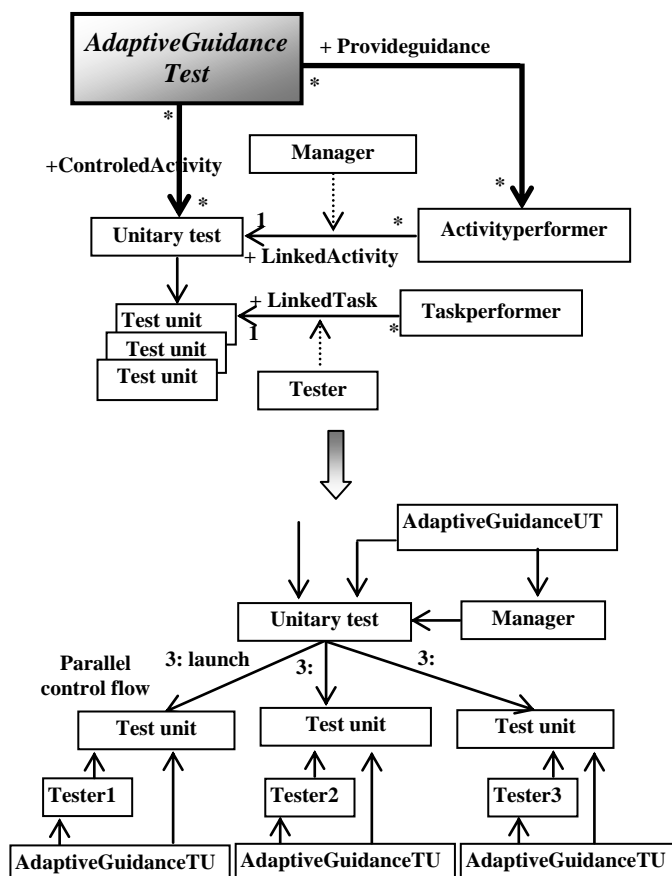


Figure 8- : The adaptive execution process «Unitary test»

The adaptive guidance is linked to the manager or to each tester according to the current context profile defined by their role, the activity model and performer's qualification. We explain this adaptive approach through the following situation; the testers have the same role "test unit" with identical activity model. However, the qualification differs from one tester to another. We consider three situations with tester's qualification defined respectively as high, medium, and low. The study case is related to launch the test unit without having all the input data, by selecting the appropriate test variables and generating the unit test report. The adaptive guidance process related to each qualification case is described as follows:

1. For a tester with high qualification: the tester starts the test unit process on the basis of the defined plan by taking its proper initiatives, the provided guidance intervention is the corrective order. The corrective intervention is provided to inform the manager of the setback and remind him of the corresponding unitary test diagram. The manager remains free to take into account the intervention.
2. For a tester with an average skill: the tester start the test unit process by applying rigorously the defined test plan, the provided guidance intervention is the constructive order. The guidance system analyzes the current context of the task, evaluates the impact and consequence of the delay caused in comparison with possible margins and offers a possible solution to the manager (solution: the guidance proposes to cancel the launch of the current test unit and generate a new execution plan according to the rate of delay and possible margins). The construction solution is not definite; it should be validated by the manager.
3. For a tester with a low qualification: the tester starts the test unit process by applying reliably the defined test plan, the provided guidance intervention is the automatic order. The guidance system analyzes the current context of the task, cancels the launch of the "test unit" task, evaluates the impact and consequence of the delay caused in comparison with the possible margins and automatically updates the execution plan of activity "unitary test".

The practical definition of the adaptive guidance type for each considered profile is deduced by a quantitative process of the characteristics in relation to the basic models (task, activity, task performer, activity performer). The considered example is processed as follows.

Each profile is semantically described in table (Table -3-). The semantics evaluation and the weighting are determined by the project manager under the specification of an ongoing project [14]. To scan the

semantics evaluation, we associate the weight corresponding to the consideration according to each attribute.

Considering the assumption retained at the definition of the example, the difference between the three context profiles is located only at the task performer model.

Table -3- : An example of the profiles evaluation

Basis Model	Features	Context profile 1	Context profile 2	Context profile 3	W_i
Task model	Complexity level	Low	Low	Low	P2
	Task type	Margin Free	Margin Free	Margin Free	P1
Task performer model	Role	No effect	Classic	Critique	P4
	Competence	High	Medium	Low	P1
	Familiarity with Process Software	Quite Acceptable	Medium	Low	P3
	Behavior for assistance	Adequate	Satisfying	Inadequate	P2
Activity model	Density of tasks in the activity	Acceptable	Acceptable	Acceptable	P1
	Complexity level	Medium	Medium	Medium	P2
Activity performer model	Skill Area Collaboration	Medium	Medium	Medium	P1
	Behavior for assistance	Adequate	Adequate	Adequate	P2

With $W_i \in [1, 5]$. Where P_i represents the computing value.

Considering the similar principle that the COCOMO model, the quantification of each profile's characteristic is on the data range [0, 2] (see Table -4-).

It is usually done through three steps, described by levels of high, medium or low contribution, applying the following rules:

1: impact of middle order.

<1: positive impact.

>1: negative impact.

Table -4- : The profile quantification

Basis Model	Features	Context profile 1	Context profile 2	Context profile 3	W_i
Task model	Complexity level	0.60	0.60	0.60	1
	Task type	1.20	1.20	1.20	2
Task performer model	Role	0.40	0.60	1.90	4
	Competence	0.25	1.00	1.80	2
	Familiarity with Process Software	0.50	1.00	1.60	3
	Behavior for assistance	0.40	0.80	1.70	1
Activity model	Density of tasks in the activity	0.80	0.80	0.80	2
	Complexity level	1.00	1.00	1.00	1
Activity performer model	Skill Area Collaboration	1.00	1.00	1.00	2
	Behavior for assistance	0.60	0.60	0.60	1

The guidance profile (GP) associated to each context profile (P_x) is based on the following formula:

$$GP(P_x) = \sum A_i W_i / 2 * \sum W_i \text{ avec } i=1 \text{ to } 10 \quad (1)$$

With: A_i the characteristic value and W_i the associated weight.

The guidance profile of each considered profile based on the GP value is given by (see Table -5-):

Table -5- : The associate guidance value

	Associate guidance profile 1	Associate guidance profile 2	Associate guidance profile 3
GP	0.321	0.431	0.681
Guidance Intervention	Corrective	Constructive	Automatic

It should be noted that the value of GP ranged from 0 to 1 and the range associated with each type of guidance is defined by the fixed limits to each guidance

type. If fixed the range of corrective guidance from 0 and 0.35 and the range of the constructive guidance from 0.36 to 0.65, we automatically associate a

corrective guidance to profile 1 and a constructive guidance to profile 2 and automatic guidance to profile3.

VI. Conclusion

Our main aim in this article is to propose and formalize such Meta Model called AGM (Adaptive Guidance Meta Model). In our contribution, we focused first in defining the principle limits of the studied PSEEs, classified according to their importance, into two categories of criteria. This study has been a base for the orientation and positioning of the proposed approach. Secondly, we concentrated our presentation on the detailed description of our conceptual Meta Model and the definition of our AGM modeling approach by introducing new concepts via the four defined basic models regarding the two considered abstraction levels.

This contribution is fulfilled by modeling an adaptive guidance system described by the integration of stereotypes related to guidance at the packages profile, "MethodContent" "ProcessWithMethod", and their description by new stereotyped classes and respective associations.

A perspective to this work concerns the development of semantic rules which allow to swap through different guidance profiles, either statically by adjustment of guidance parameters or dynamically through the performer behavior.

References

- [1] B. Combemale, A. Caplain, X. Crégut, B. Coulette. Vers une vérification d'un procédé de développement modélisé en SPEM. In: Formalisation des Activités Concurrentes, Toulouse, French 2006.
- [2] MALGOUYRES H., MOTET G, 2006. A UML Consistency Verification Approach Based on Meta modeling Formalization, Symposium on Applied Computing, Dijon, France, ACM publishers, April 2006.
- [3] Hamid Khemissa, 2008. A Generic assistance system of software process, in International Conference on Software Engineering: Software Engineering. SE 2008, Feb 12-14-2008, Innsbruck, Austria.
- [4] Hans-Ulrich Kobialka, 2004. Supporting the Software Process in A Process-centered Software Engineering environment », Upgrade-cepis.org/issues/2004/5/upgrade-v VOL; V n° 5 October 2004.
- [5] Jacky Estublier, Jorge Villalobos, Tuyet Le Anh, Sonia Jamal-Sanlavielle and German Vega. 2003. An Approach and Framework for Extensible Process Support System. In Proceedings 9th European Workshop on Software Process Technology, Helsinki, Finland, 2003-09-01.
- [6] Coulette Bernard., Crégut Xavier. & al, 2000. RHODES, a Process-centered Software Engineering Environment, in Proc. of ICEIS 2000, Stafford, pp 253-260, 2000.
- [7] Ivan Garcia and Carla Pacheco, 2009. Toward Automated Support for Software Process Improvement Initiatives in Small and Medium Size Enterprises ». Book chapter. Software Engineering Research, Management and Applications 2009 Volume 253/2009, pp. 51–58. c_ Springer-Verlag Berlin Heidelberg 2009. ISBN: 978-3-642-05440-2.
- [8] OMG. Inc. Software and System Process Engineering Meta-Model Specification version 2.0: Formal/2008-04-01.
- [9] Gruhn V., 2002. PSSEs, A Brief History and Future Challenges, Annals of S.E. 14(1-4), Kluwer, Academic Publishers, Netherlands, pp 363-382, 2002.
- [10] Tran Hanh Nhi, Bernard Coulette, Xavier Crégut, Thuy Dong Thi Bich, Thu Tran Dan, 2003. Modélisation du méta-procédé RHODES avec SPEM. Dans : Recherche Informatique Vietnam-Francophone (RIVF'03), Hanoi, Vietnam, 2003.
- [11] Pascaline Laure Tchienehom, 2005. Modèle générique de profils pour la personnalisation de l'accès à l'information. In INFORSID 2005, 24/05/2005-27/05/2005. p.269-284. Grenoble, France.
- [12] Clarke, Paul and O'Connor, Rory, 2011. *An approach to evaluating software process adaptation*. In: 11th International SPICE Conference on Process Improvement and Capability dEtermination, 30 May - 1 June 2011, Dublin, Ireland. ISBN 978-3-642-21233-8.
- [13] Grambow, Gregor and Oberhauser, Roy and Reichert, Manfred, 2011. Enabling Automatic Process-aware Collaboration Support in Software Engineering Projects. In: Selected Papers of the ICISOFT'11 Conference. Communications in Computer and Information Science (CCIS).
- [14] Barry W. Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, Bert Steece, 2009. Software Cost Estimation with COCOMO II. Prentice Hall Edition, ISBN: 0137025769, 978013702576.

The authors' profile

Mourad Chabane Oussalah is a full Professor of Computer Science at the University of Nantes and the

chief of the software architecture modeling Team. His research concerns software architecture, object architecture and their evolution. He worked on several European projects (Esprit, Ist, ...). He is (and was) the leader of national project (France Telecom, Bouygues telecom, Aker-Yard-STX, ...). He earned a BS degree in Mathematics in 1983, and Habilitation thesis from the University of Montpellier in 1992.

Mohamed Ahmes-Nacer is a full Professor at USTHB (Algiers's University). He is the Director of Computer Engineering Laboratory at USTHB and is in charge of the software engineering team. He published extensively and his current research interests include process modelling, information systems, software architecture based components and service web development.

Hamid Khemissa is teacher/researcher at Computer Systems Department, Faculty of Electronics and Computer Science, USTHB University, Algiers. He is member of the software engineering team at computer system laboratory LSI, USTHB. His interest domains concern Software Process Modeling and Software Modeling Assistance.